AN ADAPTIVE CRYPTOGRAPHY ARCHITECTURE FOR HIGHLY SECURED DATA TRANSMISSION IN IOT

A SYNOPSIS

Submitted by

SIVABALAN N
(Reg. No. 201907272)

in partial fulfillment for the award of the degree of

DOCTOR OF PHILOSOPHY

SCHOOL OF COMPUTING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

KALASALINGAM ACADEMY OF RESEARCH AND EDUCATION

(DEEMED TO BE UNIVERSITY)

ANAND NAGAR, KRISHNANKOIL – 626 126

TAMIL NADU, INDIA

MARCH 2023

Abstract

The Internet of Things (IoT) is a computing idea that enables interconnection between items with distinctive identifiers. IOT offers a wide range of applications, including e-health, wearable technology, smart homes, smart cities, etc. These factors have caused millions of devices to be interconnected. These interconnected devices will be able to gather data, analyse it, and even come to solid choices on their own. In such circumstances, security is an essential need, and authentication shows significant importance since the damage can be caused by a malicious unauthenticated device in an IoT system. Due to the heterogeneity of the linked devices in an IoT system, authentication is a difficult process. Devices can communicate with users in a secure manner due to the authentication process. System-On-Chips (SoCs) for IoT applications need hardware-based arbitrary number generators for safe data transmission. It does not have a strict hardware and they aim to employ a specific Twisted Edwards Curve (TEC). In the first paper, a symmetric encryption with integrated TEC is proposed. The Advanced Encryption Standard (AES) key extension method is developed to demonstrate the dual application of the suggested TEC. The substitute-permute network design rule is the only principle on which the AES method relies, and it works admirably in both hardware and software. The method described here uses a single indistinguishable secret key to encrypt and decrypt data.

The devices in the IoT are vulnerable to security risks, making it challenging to develop an IoT application. Therefore, a cryptographic process that is efficient must be created using the least amount of hardware resources possible. A better AES design is suggested in the second study to increase security amongst IoT devices. The suggested AES architecture incorporates the following crucial tactics: 1) An Effective Pseudo-Random Number Generator (EPRNG) that generates a unique optimal key value for each clock cycle utilising a two-level True Random Number Generator as the basis; 2) The AES architecture uses the fewest logical elements possible because automatic key creation eliminates the need for registers to store the created keys. Not all protocols are impervious to malicious attacks; some are vulnerable to side channel, DoS, and router implementation attacks. A slight vulnerability to these assaults could make it easier for the attacker to access the safeguarded data. Data integrity and confidentiality suffer as a result, and there is a tendency for a rise in computation and communication overhead.

An unauthorised user is now able to attack the information since the public key in the ECQV is slow to verify signatures. In the third research, the Two Factor Authentication (TFA-RURV-IoT) architecture is used to address the aforementioned issues. Even for the same difficulties, the Reconfigurable Unpredictable Response Value (RURV) model delivers various response values for each clock cycle. Additionally, S-box based Flipping Module (SBFM) architecture aids in server-side generation of the random seed values. This technique avoids the manual seed insertion process that increases the speed of the overall architecture and decreases design complexity.

LIST OF CONTENTS

Contents

Abstract 1

1. Introduction 4

2. Literature Review 4

3. Objective 6

4. Research Problem 6

5. Proposed Methodology 7

5.1 Proposed Method 1 7

5.2 Proposed Method 2 7

5.3 Proposed Method 3 9

6. Experimental Results and Discussion 11

6.1 Experimental Results 1 11

6.2 Experimental Results 2 11

6.3 Experimental Results 3 14

7 Organization of Thesis 18

8 Conclusion 18

9 Reference 20

10 List of Publications 21

1. Introduction

The Internet of Things (IoT) introduces a new realm in which all gadgets and products are linked to one another through a worldwide internet and may be used collectively by the user to carry out all complex tasks effectively. The Internet of Things (IoT) is expanding with the invention of numerous new gadgets, such as the Radio Frequency Identification Device (RFID), sensors and others [1]. In regards to social life equipment like smart grid, smart homes, communication, fitness, surveillance systems, security, entertainment, medical, etc., IoT is now a part of everyone's daily life. IoT security is crucial since privacy protection is more critical than ever since malicious individuals can simply target IoT devices and cause the whole system to a rest [2]. Dyn, a DNS service provider was the target of a serious Distributed Denial of Service (DDoS) attack that had a significant impact on well-known social media Platforms like twitter, Amazon, and Linkedin [3].

IoT system needs to be extremely protected against unauthorised access since, after successfully breaching one layer of the IoT, attackers can simply take full control of the system through the compromised node. In addition, viruses, hackers, and malicious software may corrupt the system and data integrity, posing a serious threat to the IoT environment. The IoT's Security domain has enormous potential. The IoT architecture of IoT has a lot of unresolved problems. Wireless Sensor Networks (WSN) and IP-based WSNs are easy targets for hackers and present a significant security issue in IoT architecture [4]. The entire Internet of Things system might be brought to a standstill if the hacker gained access to even one node. The elements in every IoT system are dispersed geographically, and wireless communication is necessary for them to connect with one another. It is far more difficult to deal with threats and exploitation in an IoT context since wireless networks are more susceptible to them. Data absorption for practical systems like traffic analysis and quick reaction systems should be swift and precise [5].

2. Literature Review

The PN Sequence Generator (PNSG) was introduced by Zodpe and Sapkal [6] to produce the S-box. The initial seed value and feedback taps were employed to generate a unique set of random integers that use the PSNG. The PNSG's properties were utilised to boost the cryptosystem's effectiveness. Since the AES currently

provides a mechanism for generating initial keys, no extra key has been needed. However, the non-pipelined stage design of AES required a lot of hardware resources to implement.

The Lightweight AES (LAES) method was developed by Kumar et al. [7] which has been developed in Kintex-7 and Artix-7 FPGAs for voice data security. The LAES algorithm lowered the MixColumns operation, resulting in a reduction in delay. Additionally, the logic operation was reduced by the lesser MixColumns procedure. As a result, the difficulty of the LAES was decreased during the voice data encryption. However, reducing the MixColumns function enhanced the use of multipliers.

The high throughput FPGA architecture of the AES was created for automotive industries by Sikka et al. [8]. The Vivado High-Level Synthesis (HLS) tool was used to create the 128-bit key, 10 block size AES. The application-specific bit widths used to create the FPGA were dependent on the HLS. Additionally, by employing the HLS, the AES's speed was increased. However, the constant recalibration of the signal length caused AES to have too much delay.

The Modified Mix Column (MMC) with gate replacement was created by Madhavapandian and MaruthuPandi [9] to construct a stable structure of AES for internet control message protocol technology. Therefore, the AES was created using a resource-sharing design and an effective mix column Boolean expression. By improving the entire framework, the AES architecture's power was reduced. However, because it increased communication overhead, it was necessary to reduce the computational complexity.

A secure multi-cloud architecture was presented by Megouache et al. [10] to enhance the data's authenticity and secrecy. Three separate methods—Virtual Private Network (VPN), RSA, and hashing algorithm were used in the established secure cloud. At first, RSA was employed to offer the data security while a VPN was established between the supplier and the consumer. Additionally, the validation was provided by the hashing function that ensured the gathering of data from the clouds. Here, the VPN was utilised to lower the possibility of data loss while communicating. Conversely, DDoS attacks could be launched against the built multi-cloud architecture.

3. Objective

- To offer security for response values, EPRNG-AES architecture with 128-bit input, key, and cipher text values are proposed. For every clock cycle, the EPRNG with a two-level TRNG module generates a unique key value.
- To enhance the resilience against unauthorized users, Different key values and response values are established with more dissimilarity among the encrypted values. As a result, only authorized individuals can access the data sent between IoT devices and servers.
- To prevent side channel attacks, Each Internet of Things device requests the server for a specific challenge value, Master Key (MK), and Alias Identify Key (AID). These distinctive values aid in protecting the data from adversarial models.
- To produce various response values for each clock cycle, the Reconfigurable Unpredictable Response Value (RURV) mechanism is used.
- To produce a unique seed for each clock cycle, S-box based Flipping Module (SBFM) is used to generate the seed values on the server side.

4. Research Problem

- Since it analyses a significant volume of data during communication, collision happens. Whenever two devices collide, the communication system experiences data loss.
- The non-pipelined stages used in the development of the AES design result in higher hardware consumption.
- In addition, the function of MixColumns is reduced while the use of multiplexers is raised. The signal width is frequently recalculated, increasing the delay that reduces the operating frequency.
- Privacy concerns are being brought about by the growing number of IoT devices. The effectiveness of the design model is commonly reduced by side channel and brute force attacks on the system model.
- Because of the significant increase in the rate at which data is accumulating, any IoT device may have to contend with the scaling problem. Each clock cycle in the server side requires the manual seed values that increases the architecture's difficulty and delay.

5. Proposed Methodology

5.1 Proposed Method 1

This part presents the turned Edwards curve and associated gathering regulation. An efficient method of iterative midway item expansion is suggested for modular multiplication. Figure 1 shows the suggested modular multiplier over GF(p) in the integral of this method. In this process, gatherer C is increased at the starting of every cycle to enable iterative enlargement of the continuous halfway items. In order to put as much focus on repeated operation as feasible from the beginning of power usage, PD and PA modules entirely absorb power. While point expansion has a longer delay than point multiplication, point multiplying is performed prior to point expansion. Mostly the evaluation plan on multiplication point wise calculations has been consumed by Module PA. The piece sample of the mystery key cannot be evaluated using the comparative power design.

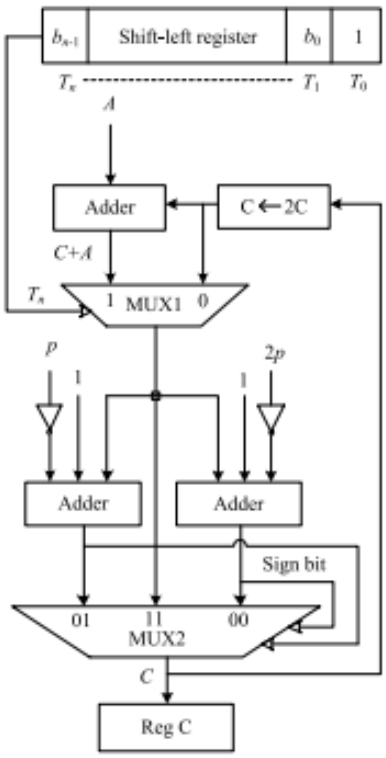

Figure 1. Architecture diagram

5.2 Proposed Method 2

The EPRNG-AES architecture uses optimum key generation to improve the protection of IoT devices. The modified AES architecture is used to secure the response value produced for the IoT device. Furthermore, the challenge value is constant across

all clock cycles, while this answer value varies for each cycle. The Ciphertext differ greatly from one another as a result, increasing the security of the communication made by IoT devices. In contrast, by improving the AES architecture, the hardware consumption of the EPRNG-AES architecture is reduced. The two-level TRNG is utilised as input in the EPRNG architecture to provide an ideal key value of 128 bits for enhancing security against various attacks. EPRNG is the name given to the overall key generation procedure that employs combinational blocks and a two-level TRNG. For each clock cycle, this EPRNG produces a distinct integer, or key. The two tiers of the TRNG units are often used in this instance to create a fresh number which can be used as input to create the best key. This newly created EPRNG key generation module, which is seen in Figure 2, is utilised to produce 20-bit key values. In order to create six 20-bit key values, six EPRNG key generation modules are utilised, yielding 120-bit key values. The key values are then merged in order to carry out the encryption and decryption operation.

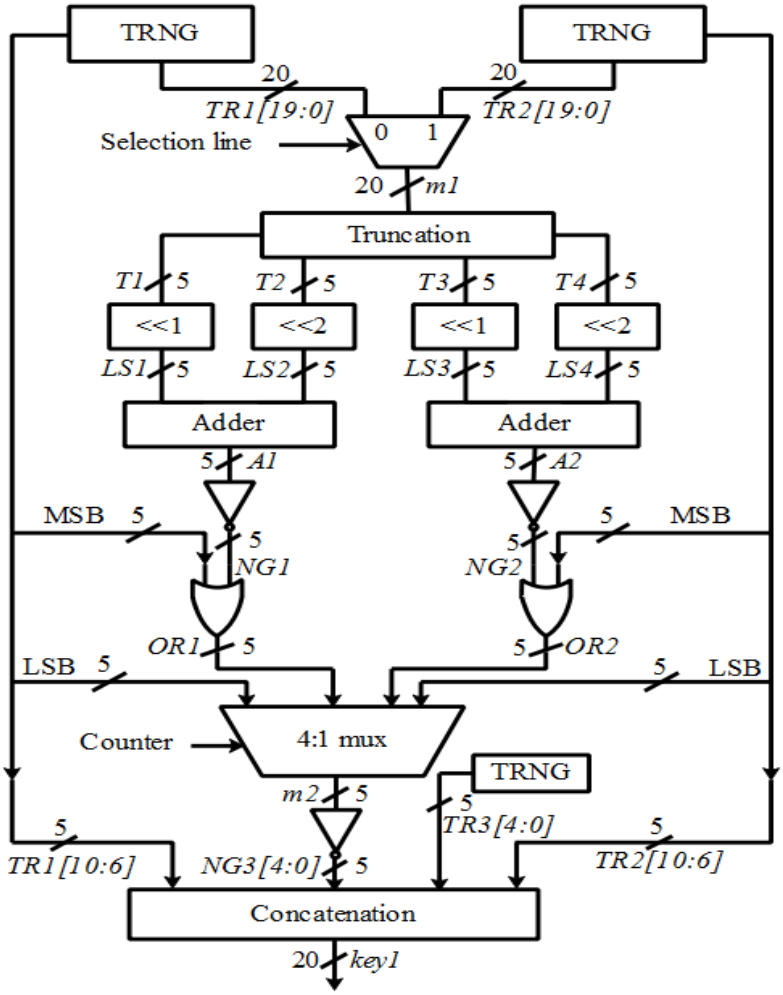

Figure 2. EPRNG key generation architecture

Every clock cycle is given a unique key value by the configured EPRNG, and this key value is utilised to secure the IoT device's response. The server generates the server nonce and the hash key response during the authentication process. The EPRNG-AES architecture uses a minimum number of logical elements by not using registers to maintain the generated keys. The server then gives the IoT device the authorisation required to allow data transmission among the IoT device and server. Because the encrypted values are varied when using a different key and response values, the robustness against unauthorised users is maximised. As a result, the server and Internet of Things device are more securely connected with EPRNG-AES architecture, which also reduces the logical components of AES.

5.3 Proposed Method 3

Although there are still some restrictions, all of the current authentication systems are generally successful. It may be inferred from their performance study that those strategies drain the network's resources and add computational cost. It is advised to select a lightweight authentication protocol with Unclonable Function Key (UFK) to enhance the functionality of sensor devices in the system in order to avoid these drawbacks of employing sophisticated authentication protocols.

5.3.1 Reconfigurable URV IoT set up phase

The IoT device creates a request and an identifier that are associated with the server during setup. Figure 3 depicts the setup phase's flowchart. The response was produced using the reconfigurable URV that is described in more detail in the subsection below. The server stores the created response values, from which the one-time Alias Identity, Master Key, Fake Identity, and synchronisation Key are formed. These values are all saved in the corresponding IoT devices.

5.3.2 Authentication phase

Access must be granted to the ideal IoT devices at the authentication phase, in which the server and device slot are in sync. The AID has been verified by the random number request during authentication, and message (M1) has been generated for first acceptance. A master key, challenge value, and answer have been retrieved if the AID matches the MI request; alternatively, the request has been refused. All of the message values are combined in the server's response's server nonce and hash key (M2). The IoT device created the updated M3 values if the response matches. The Fuzzy Extractor

recovery model uses the produced M3 values. The reconfigurable URV generates a random number that is verified to the key hash answer on each clock cycle. IoT device and server blocks are distinguished by various colours. In the last stage of the authentication procedure, all calculated values are modified.

5.3.3 Reconfigurable URV

The challenge is typically generated by IoT devices and sent to the server for user identification. Typically, an IoT device generates a random number for security reasons. Previous versions gave the identical response for each clock cycle, making it too simple for someone without authorization to get the data. Therefore, in this study work, a reconfigurable URV random number was already developed.

5.3.4 S-box based flipping module

To increase the unpredictability of the AID, challenge, device nonce, and server nonce from the SBFM, an 8-bit seed value is produced on the server side. The generation of random keys by the SBFM is used to increase the security of device-to-server communication. Figure 3 depicts the SBFM's design.

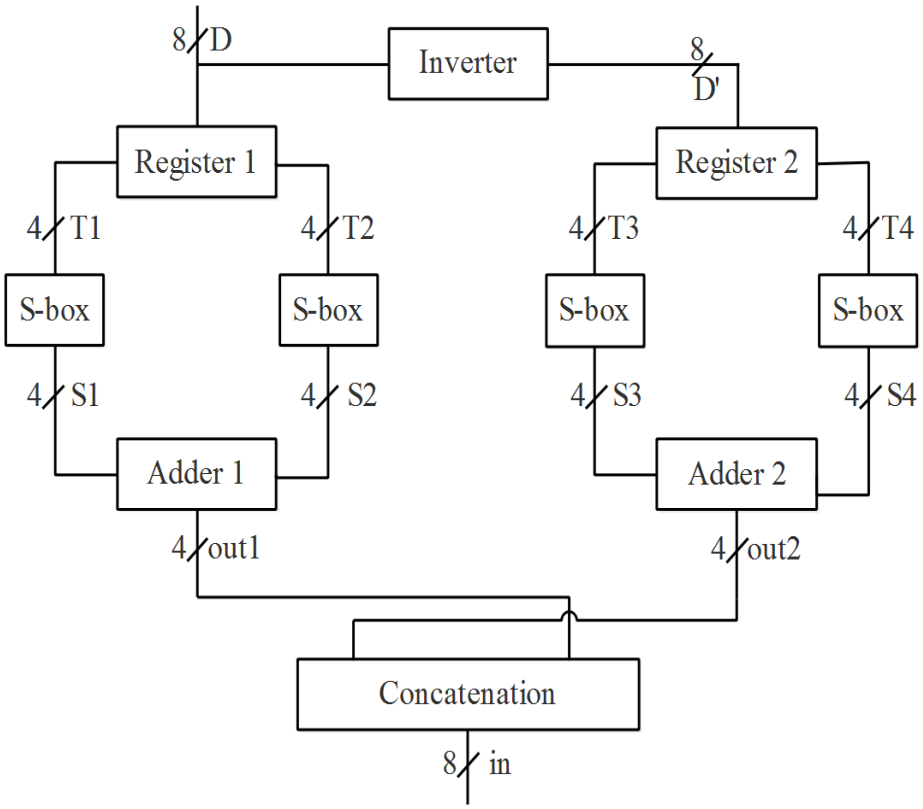

Figure 3. S-box based flipping module design

6. Experimental Results and Discussion

6.1 Experimental Results 1

The Xilinx ISE 14.7 Design Suite is used to program the processor that uses ECC, and the Xilinx ISim simulation environment is used for testing. The simulation achievements are supported by the Maple program. The results of the suggested ECC execution units over a broader field of counting up to 256 pieces are tallied. The PA, PD, and TEC modules of the suggested modulation-based multiplier operate at a maximum recurrence of 170 MHz on the Virtex-7 FPGA. The respective latencies of the TEC, PD, PA, and multiplier modules are 257, 1029, 516, and 262,650 clock cycles.

Table 1. Performance Analysis

Operations	Clock Cycles	Number of Slices	Number of LUTs	Time (µs)	Throughput (Mbps)
Point Doubling	516	1990	6958	3.2	79.93
Point addition	1029	4339	15489	3.9	40.08
Modular Multiplication	257	414	1310	1.6	160.47
Proposed Method	262650	26259	33238	1.3	15.702

A comparison of the suggested TEC plan with various available TEC over Fp-256 configurations is shown in Table 1. Their processor provides quick registering at the expense of additional hardware resources, making it unusable for devices with limited resources. The processor mentioned is SPA-safe and can be changed to accommodate different field orders. The processor uses 14. DSP cuts and 26.25k cuts on a Virtex-4 FPGA to perform TEC in 29.84 milliseconds with an 8.58 kbps throughput. The suggested processor outperforms the CPU in terms of execution speed and throughput. The main goal is to carry out TEC work on keys of number 21 simultaneously while maintaining higher throughput (15.702 mbps).

6.2 Experimental Results 2

Utilizing the Xilinx 14.4 software, the suggested EPRNG-AES architecture has been built, and the system is run on a 500 GB hard drive and 4 GB of RAM. The EPRNG-AES architecture's hardware consumption is analysed using this Xilinx

programme. Furthermore, the EPRNG-AES architecture is analysed utilising the simulation output using Modelsim 10.5 programme. The 128-bit input, key, and output data are processed using the intended EPRNG-AES architecture. Utilizing four distinct FPGA devices—Kintex 7, Virtex 5, Virtex 6, and Spartan 6—the performance of the EPRNG-AES architecture is examined.

Table 2. Performance analysis of various FPGA devices

FPGA devices	Power (W)	Delay (ns)	Operating frequency (MHz)
Kintex 7	0.038	4.383	305.761
Virtex 5	0.010	1.094	658.288
Virtex 6	0.022	2.981	266.902
Spartan 6	0.014	1.986	301.231

In addition, table 2 provides an analysis of the power, latency, and operating frequency for the EPRNG-AES design. For proposed design, the hardware consumption is examined using 128-bit input, key, and cypher text. Slices, LUTs, registers, flip flops, and logical elements make up 1-3% of the resources used by the planned EPRNG-AES architecture.

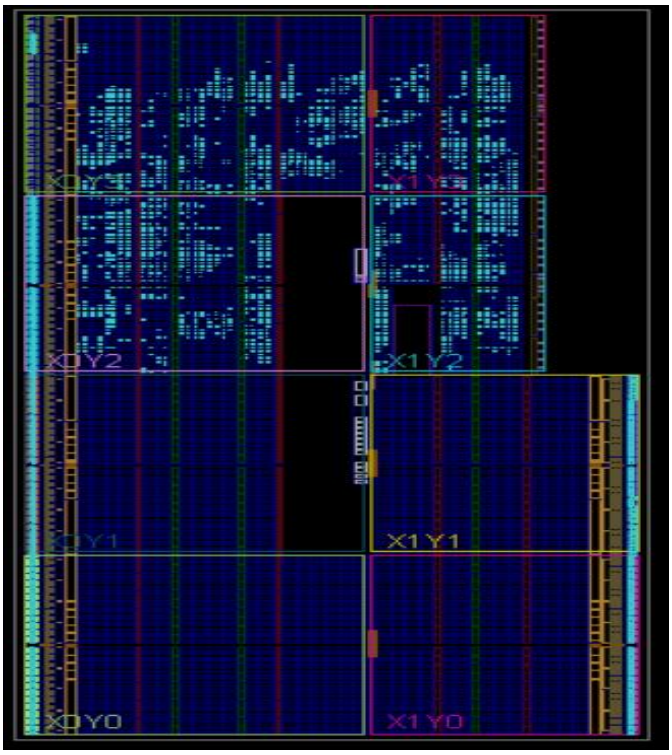

Figure 4. Floor planning design for EPRNG-AES architecture

The floor planning design for the EPRNG-AES architecture is shown in Figure 4. Utilizing floor planning concept, the EPRNG-AES architecture is validated to be connected accurately. Meanwhile, this floor planning output is only usable whenever the connectivity of the EPRNG-AES design are flawless. As a result, there are no errors in the EPRNG-AES architectural connections.

6.2.1 Security analysis of the EPRNG-AES architecture

Furthermore, the system evaluation for the LAKD, RSA, SMAP, and EPRNG-AES is assessed in terms of security, intricacy, cost, and authenticity. This assessment is displayed in Table 3 below.

Table 3. System estimation Analysis

Design	Security (%)	Complexity (%)	Cost (%)	Integrity (%)
LAKD	88	8	13	88
RSA	85	10	15	85
SMAP	74	24	36	71
EPRNG-AES	93	5	10	92

6.2.2 Comparative analysis

The quantitative examination of the EPRNG-AES architecture is provided in this study. Five current AES architectures are contrasted with the suggested EPRNG-AES architecture. Four distinct FPGA devices, including the Spartan 6, Kintex 7, Virtex 5, and Virtex 6, are compared in this instance which is shows in Table 4, 5, 6 and 7.

Table 4. Analysis for Spartan 6

Methods	Slices	Operating frequency (MHz)
AES-PNSG	5566	237.45
EPRNG-AES	55	301.231

Table 5. Analysis for Kintex 7

Methods	Slice LUTs	Flip flops	IOBs	Operating frequency (MHz)
LAES	9468	-	384	-

AES-HLS	577	449	265	297.3
EPRNG-AES	153	39	26	305.761

Table 6. Analysis for Virtex 5

Methods	**Slice registers**	**Slice LUTs**	**Slices**	**Operating frequency (MHz)**
AES-CTR	19123	14966	5974	622.4
EPRNG-AES	38	174	62	658.288

Table 7. Analysis for Virtex 6

Methods	**Slice registers**	**Slice LUTs**	**Delay (ns)**	**Power (W)**
AES-MMC	2688	9393	3.167	3.725
EPRNG-AES	38	229	2.981	0.022

The investigation leads to the conclusion that the EPRNG-AES architecture performs superior to the current one. The keys generated from the manual key generation must be stored in a large number of registers in the conventional AES designs. The current architecture needs a lot of logical pieces if there are a lot of registers. The EPRNG-AES design only needs fewer registers, though, as this improved AES makes use of the EPRNG key generation which has fewer logical components because this EPRNG key generation module doesn't need a register to store the key values. Using the EPRNG module also enhances the security of the AES design. The EPRNG module enhances the security by generating a fresh key for every clock cycle, thwarting threats like SCA, DoS, OPGA, and SKA.

6.3 Experimental Results 3

The setup for the suggested TFA-RURV-IoT includes 4 GB of RAM and a 500 GB hard drive. Verilog was used to construct the authentication and setup step that contributed with the logical element design. The hardware consumption of the suggested technique has been assessed and validated with the provisions made using Xilinx 14.4 software. The simulation that aids in confirming the authenticity of the phase and configuration mode was performed using the Modelsim 10.5 software.

6.3.1 Hardware utilization and authentication performances

The hardware consumption of the current architecture and the suggested architecture is described in detail in this article. The results are summarised in Table 8 for both architectures, which are executed in the Verilog programming language. Consequently, the reconfigurable URV use, as is depicted in CGPS Table 8, meant that the hardware requirements of TFA design were lower. Table 9 compares the suggested RURV authentication architecture to the standard IoT URV authentication.

Table 8. Analysis of hardware utilization for a different architecture

Target Device	Module	Delay	Slice	Frequency (MHz)	LUT	Flip flop
Spartan 6	TFA-URV-IoT (Gope & Sikdar 2018)	23ns	1628	250.5	108	749
	TFA-RURV- IoT	13ns	1375	460	96	640

Table 9. Analysis of hardware utilization for conventional and proposed URV model

Target device	Models	Flip flop	Frequency (MHz)	LUT	Slice	Delay
Spartan 6	URV	8	663	11	8	1.507ns
	RURV	2	678	8	8	1.473ns

According to a reference of hardware utilisation overall, the RURV-IoT uses 15.54% fewer slices, 11.11% fewer LUTs, and 14.55% fewer flip flops than the URV-IoT. The suggested solution also enhanced operating frequency by 45.54% that is a measure of system speed. Additionally, Table 8 provides the independent module findings for the URV and RURV modules. In comparison to the present URV, the suggested URV focused on in the security measures, and the outcomes are validated. For URV-IoT and RURV-IoT in both instances, the random response for each clock cycle variable has been examined. The suggested RURV-IoT architecture simply generates random data for each clock cycle, and the identical number of inputs are provided in the following clock cycle, in contrast to all traditional techniques. This unpredictability prevents unauthorised individuals from accessing private data stored on IoT devices.

Table 10. Execution time analysis

Design	Execution time (uS)
SHA- 256	37.35
AES – 256	34.26
DCC	10
DIMC	30
EBSG	256.4
LAP-M2MC	26.24
TFA-RURV- IoT	7.52

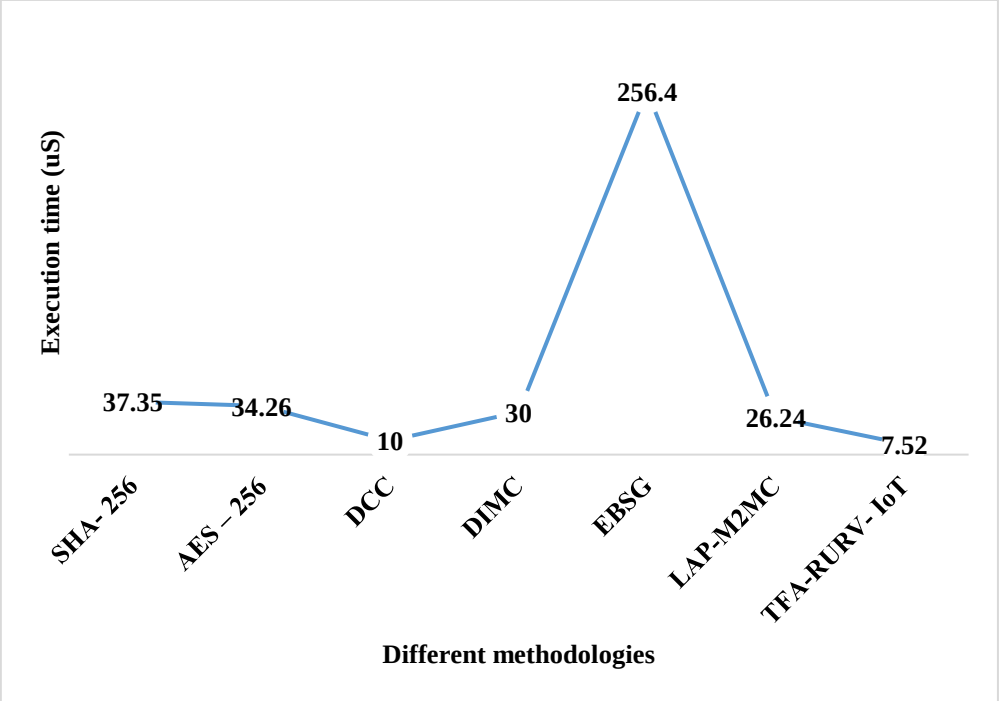

Figure 5. Graphical illustration of execution time for different methodologies

This part identifies the developed model TFA-RURV-execution IoT's time to those of other methods already in use. The results are shown in Table 10 and the corresponding comparison graph is displayed in Figure 5, respectively. However in this case, database deterioration causes a collision problem, which extends the execution time by roughly 37.35 us. When this is going on, the Advanced Encryption Standard (AES) computation encryption speed suffers when used in CBC mode because sequential block encryption lengthens execution time by approximately 34.26us. In contrast, the Distributed Cloud Computing (DCC) technique requires 10us of execution

time, and this is significantly less since it divides the complex data into smaller chunks before concurrently encrypting. Additionally, Data Integrity Multi Cloud (DIMC) parallelized the encryption process and achieved an execution time of 30us by dispersing the data across various cloud systems. Additionally, even though the protocol is lightweight for authentication, the computation time for the Edge Based Smart Grid (EBSG) scheme is 256.4us, which is significantly longer. Since the hash function cannot perform as expected, the database set degrades quickly, that tends to increase the collision rate. Lightweight Authentication Protocol for Machine to Machine Communication (LAP-M2MC) finally achieved computation time of about 26.24us in the IIoT environment, which is a little bit faster than conventional methods. The recommended design's execution took 7.52 us because RURV and SBFM were used. Additionally, Table 11 and Figure 6 demonstrate how the system estimation compares to one another. Comparing the conceptual model to existing systems, the security, complexity, cost, and integrity have all increased.

Table 11. Comparison of system estimation

Design	Security (%)	Complexity (%)	Cost (%)	Integrity (%)
SHA- 256	80	15	25	78
LAP-M2MC	88	8	13	88
EBSG	74	24	36	71
DIMC	85	10	15	85
DCC	86	8	14	86
AES – 256	81	18	29	81
TFA-RURV- IoT	91	6	11	89

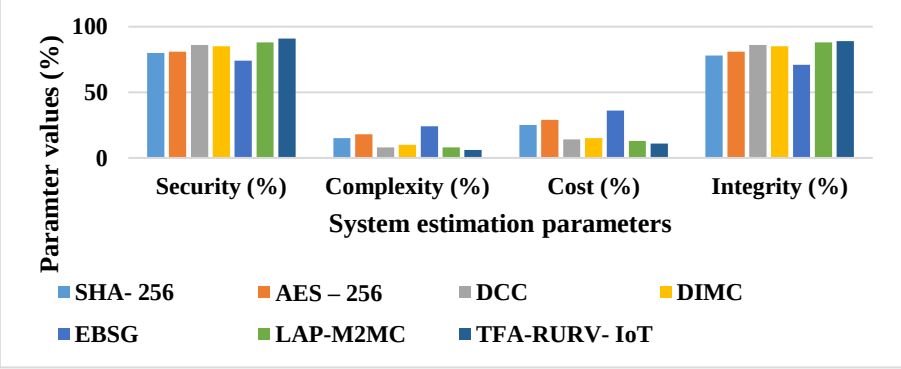

Figure 6. Comparison graph of system estimation

7. Organization of Thesis

The organization of the thesis is mentioned as follows,

Chapter 1 provides the detailed description about low power FPGA implementation, security, lightweight authentication, procedure, challenges, objectives, design application, and research goals.

In Chapter 2, the merits and disadvantages of the Low Area FPGA Implementation and Security Analysis of existing researches are surveyed.

In chapter 3, An High Speed Area Efficient Twisted Edwards Curve Point Multiplication Using FPGA Architecture is described. The problem description for the current approaches is outlined, as well as the contribution of the suggested architecture. Additionally, an evaluation with the current design is given.

Chapter 4 explains two factor authentication models for various network contexts to increase data security. In order to generate various response values for each clock cycle, the Reconfigurable Unpredictable Response Value (RURV) method is employed. The MCRM-CLA-contribution and the problem statement are provided. The analytical study is assessed and contrasted with the currently used approach.

Chapter 5 explained Low Area FPGA Implementation of the AES Architecture with EPRNG for IoT Applications. The contribution of the suggested design is described, along with the problem statement for the current approach of noise reduction. The effectiveness of the suggested design is assessed and contrasted with the current approach.

Chapter 6 gives the thesis's conclusion as well as the research's future work.

8. Conclusion

Security of the Internet of Things (IoT) is an active study topic in academia and the research community. Investigating various IoT security vulnerabilities will require more thought and research. With relation to IoT, the security concerns in the support layer have not yet been looked into. In Edwards25519 curve and its projective depiction, a fast, region-effective, and SCA-safe ECC processor for speedy point multiplication is proposed. For the objective of modular multiplication, a modular

operation multiplier with a radix-2 interleave takes two numbers of n bits for duplication with n+1 cycles of clock. For TEC, the Montgomery scalar multiplication approach is employed since it is quick and offers a significant advantage for SCAs. It improves throughput efficiency and region delay items without sacrificing security.

In the second research article, the EPRNG-AES design reduces hardware requirements while enhancing security among IoT devices and servers. Specific key values are generated by the EPRNG with a two-level TRNG key generation module for every clock cycle, which enhances the security of the response value. In contrast, by enhancing the AES architecture, the hardware resources are reduced. Here, the intermediate key value register is not needed for the encryption and decryption processes. As a result, the execution of the EPRNG-AES architecture uses less hardware resources and number of slices. To protect sensitive information and device-to-device connectivity, IoT device authentication is crucial. The LFSR circuit model has been used to develop a reconfigurable URV in the third research. For each clock cycle in the LFSR circuit, the feedback connection has been perceived differently in the registers. The two-factor authentication is carried out using unpredictable random data that is generated using a reconfigurable process. Depending on the key hash function results, the IoT device requests authentication from the server. The suggested TFA-RURV-IoT reduces 15.54% of Slices, 11.11% of LUT, and 14.55% of Flip flips. Furthermore, the suggested approach boosted operating frequency by 45.54%, which represents the speed of the overall system.

9. Reference

[1] R. De Smet, T. Vandervelden, K. Steenhaut and A. Braeken, Lightweight PUF based authentication scheme for fog architecture, Wireless Networks, **27** (2) (2021), 947 – 959.

[2] M. Almulhim, N. Islam and N. Zaman, A lightweight and secure authentication scheme for IoT based e-health applications, International Journal of Computer Science and Network Security, **19** (1) (2019), 107 – 120.

[3] S.S. Dhanda, B. Singh and P. Jindal, Lightweight Cryptography: A Solution to Secure IoT, Wireless Personal Communications, **112** (3) (2020), 1947 – 1980.

[4] J. Zhang, C. Shen, H. Su, M.T. Arafin and G. Qu, Voltage over-scaling-based lightweight authentication for IoT security, IEEE Transactions on Computers, **71** (2) (2021), 323 – 336.

[5] S. Amanlou, M.K. Hasan and K.A.A. Bakar, Lightweight and secure authentication scheme for IoT network based on publish–subscribe fog computing model, Computer Networks, **199** (2021), 108465.

[6] H. Zodpe and A. Sapkal, An efcient AES implementation using FPGA with enhanced security features, Journal of King Saud University, **32** (2) (2020), 115 – 122.

[7] K. Kumar, K.R. Ramkumar, A. Kaur, A lightweight AES algorithm implementation for encrypting voice messages using field programmable gate arrays, Journal of King Saud University-Computer and Information Sciences, **34** (6) (2022), 3878 – 3885.

[8] P. Sikka, A.R. Asati, C. Shekhar, RETRACTED ARTICLE: High-throughput field programmable gate array implementation of the advanced encryption standard algorithm for automotive security applications, Journal of Ambient Intelligence and Humanized Computing, **12** (7) (2021), 7273 – 7279.

[9] S. Madhavapandian and P. MaruthuPandi, FPGA implementation of highly scalable AES algorithm using modified mix column with gate replacement technique for security application in TCP/ IP, Microprocessors and Microsystems, **73** (2020), 102972.

[10] L. Megouache, A. Zitouni and M. Djoudi, Ensuring user authentication and data integrity in multi-cloud environment, Human-centric Computing and Information Sciences, **10** (2020), 15.

10. List of Publications

1. N. S. Balan and B. S. Murugan, An High Speed Area Efficient Implementation of Prime Field based Twisted Edwards Curve Point Multiplication using FPGA Architecture, 2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics, ICDCECE, (2022), 1-5.

2. N. Sivabalan and B.S. Murugan, Low Area FPGA Implementation of AES Architecture with EPRNG for IoT Application, Journal of Electronic Testing, **1** (2022), 181-193

3. N. Sivabalan and B.S. Murugan, Lightweight Two-Factor Authentication With S-Box Flipping Module for IoT Security, International Journal of e-Collaboration (IJeC), **18**(1) (2022), 1 – 24.